# Data Set Information Sheet: Pushing the Limits of Remote RF Sensing: Reading Lips Under Face Mask

This “ReadMe.pdf” file was generated on 2022-01-30 by Hira Hameed

## GENERAL INFORMATION

### Title of Dataset

### Pushing the Limits of Remote RF Sensing: Reading Lips Under Face Mask

### Author Information

Names: Hira Hameed, Dr. Muhammad Usman, Dr. Ahsen Tahir, Dr. Hasan Abbas, Professor Muhammad Ali Imran, and Dr. Qammer Abbasi

Institution: University of Glasgow

Address: James Watt School of Engineering, College of Science and Engineering, University of Glasgow, Glasgow, G12 8QQ

Emails:2683961H@student.gla.ac.uk;[Muhammad.Usman@glasgow.ac.uk](mailto:Muhammad.Usman@glasgow.ac.uk); [Ahsen.Tahir@glasgow.ack](mailto:Ahsen.Tahir@glasgow.ack), [A.hussain@napier.ac.uk](mailto:A.hussain@napier.ac.uk); [Hasan.Abbas@glasgow.ac.uk](mailto:Hasan.Abbas@glasgow.ac.uk); [tjcui@seu.edu.cn](mailto:tjcui@seu.edu.cn) [Qammer.Abbasi@glasgow.ac.uk](mailto:Qammer.Abbasi@glasgow.ac.uk);  [Muhammad.Imran@glasgow.ac.uk](mailto:Muhammad.Imran@glasgow.ac.uk)

### Date of data collection:

The data was collected over the first week of September 2021.

### The geographic location of data collection:

Scotland 5G Centre, James Watt South Building, Glasgow, G12 8QQ, United Kingdom

### Information about funding sources that supported the collection of the data:

### This work was supported in parts by Engineering and Physical Sciences Research Council (EPSRC) grants: EP/T021020/1 and EP/T021063/1.

## SHARING/ACCESS INFORMATION

### Licenses/restrictions placed on the data:

NA

### Links to other publicly accessible locations of the data:

NA

### Was data derived from another source?

No

Recommended citation for this dataset: Hameed, H., Usman, M., Tahir, A., Hussain, A., Abbas, H., Cui, TJ., Imran, M.  and Abbasi, Q.  (2022), Pushing the Limits of Remote RF Sensing: Reading Lips Under Face Mask, University of Glasgow <http://dx.doi.org/10.5525/gla.researchdata.1282>

## DATA & FILE OVERVIEW

### Details of Data Folders and Files

## USRP\_Dataset

The main folder USRP\_Datset (See Figure 1). Totals have 1800 samples (see table 1).

The USRP Dataset folders have been separated into six new folders (see Figure 2). Each folder is divided into six categories: A, E, I, O, U, and Empty (see Figure 3).

Graphical user interface, application, Word

Description automatically generated*Figure 1 Data folder Structure*

Graphical user interface, application, table

Description automatically generated

Figure 2 USRP Data folder Structure

Graphical user interface, application, table, Excel

Description automatically generated

Figure 3 Each Data folder Structure

Table Details of the Data Set (Folders, Files, Description, and Number of Samples)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number of Subjects** | **Folder Name** | **Technology** | **Description** | **Number of Samples per Class** |
| 1 | Subject1\_Female1\_With Mask | USRP | The Male subject pronounced the vowel and Empty Data using Mask | 300 |
| Subject1\_Female1\_Without Mask | The Male subject pronounced the vowel and Empty Data using Without Mask | 300 |
| 2 | Subject2\_Female2\_With Mask | USRP | The female subject pronounced the vowel and Empty Data using Mask | 300 |
| Subject2\_Female1\_Without Mask | The female subject pronounced the vowel and Empty Data using Without Mask | 300 |
| 3 | Subject3\_Male1\_With Mask | USRP | The female subject pronounced the vowel and Empty Data using Mask | 300 |
| Subject3\_Male1\_Without Mask | The female subject pronounced the vowel and Empty Data using Without Mask | 300 |

**METHODOLOGICAL INFORMATION**

1. **Experimental Setup Using USRP**

### Description of methods used for collection/generation of data:

The dataset represents a combination of activities captured through wireless channel state information, using one USRP X300 devices each equipped with the VERT2450 omnidirectional antenna and HyperLOG 7040 X. HyperLOG 7040 X was used as the transmitter, and the USRP (X300) was used for the receiver. USRP and HyperLOG 7040 X were connected to an All-in-One PC that uses an Intel(R) Core (TM) i7-7700 3.60 GHz processor and 16 GB RAM. The system made use of virtual machines to provide the Ubuntu 16.04 operating system. On the Ubuntu virtual machines, Gnu Radio was used to communicate with the USRP devices, and it also hosted the artificial intelligence engine. (See Figure 4). shows the experimental setup which was used to collect the data.

Graphical user interface, diagram, website

Description automatically generated

Figure 4 Experimental Setup

### Methods for processing the data:

Firstly, the USRP receiver and HyperLOG Directional antenna as receiver were configured to communicate together using the GNU radio python package to set parameters such as central frequency, number of Orthogonal Frequency Division Multiplexing (OFDM) subcarriers, and power levels (see Table 2).

|  |  |
| --- | --- |
| Parameter | Value |
| USRP Platform | X300 |
| OFDM subcarriers | 51 |
| Operating Frequency | 2.45 GHz |
| Transmitter Gain | 35 dB |
| Receiver Gain | 35 dB |

Table System Parameters

GNU Radio is free and open-source software that is used in research for software-defined radios and signal processing. GNU Radio comes with examples of OFDM signal processing where the channel state information can be extracted. This example is modified to include the USRP as the transmitting and receiving devices. The GNU Radio software publishes the configuration in the format of a flow diagram which can be used to set up the blocks of the USRP and OFDM communication. (See Figure 5) the GNU Radio flow diagram depicts the configurations of the USRP devices. The flow diagram can then be converted into a python script, which can be executed to begin OFDM communication. The raw form of the channel state information output is written to text files which are then converted to a processable format (see Figure 6).

A picture containing diagram

Description automatically generated

Diagram, schematic

Description automatically generated

Figure 5 GNU Radio Flow Diagram

Diagram

Description automatically generated

Figure 6 Data flow in the data collection stage

Secondly, the task was to collect the channel state information and create data sets from them in the form of “Comma Separated Value” (.csv) files. The CSV files would hold the data sets that will be used for training and testing of the Machine Learning (ML) algorithm. For this, another python script is used to process the terminal output and filter out the channel state information complex numbers. Python carries out mathematical functions to calculate the amplitude of the RF signal from the channel state information complex numbers. The amplitude values are then saved to CSV format for ML and to visualise the signal propagation through line graphs, see an example in Figure 5. The above process was repeated for all the data files in this data set.

Graphical user interface, chart

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure Channel state information capture of “A” Vowel in ".csv" format and the corresponding plot

### Instrument- or software-specific information needed to interpret the data:

Data files are all in “.csv” format which could be opened using Microsoft Excel and further processed using python.

## Radar\_Dataset

The main folder name is Radar\_Datset (See Figure 1) and have total 1800 samples (see table 1).

The Radar Dataset folders have been separated into six new folders (see Figure 2). Each folder is divided into six categories: A, E, I, O, U, and Empty (see Figure 3).

Graphical user interface, application

Description automatically generated

Figure 1 Data folder Structure

Graphical user interface, application

Description automatically generated

Figure 2 Radar Data folder Structure

Graphical user interface, application

Description automatically generated

Figure 3 Each Data folder Structure

Table 1 Details of the Data Set (Folders, Files, Description, and Number of Samples)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number of Subjects** | **Folder Name** | **Technology** | **Description** | **Number of Samples per Class** |
| 1 | Subject1(M)\_With Mask | Radar | Male subject pronounced the vowel and Empty Data using Mask | 300 |
| Subject1(M)\_Without Mask | Male subject pronounced the vowel and Empty Data using Without Mask | 300 |
| 2 | Subject2(F1) \_With Mask | Radar | Female subject pronounced the vowel and Empty Data using Mask | 300 |
| Subject2(F1) \_Without Mask | Female subject pronounced the vowel and Empty Data using Without Mask | 300 |
| 3 | Subject2(F2) \_With Mask | Radar | Female subject pronounced the vowel and Empty Data using Mask | 300 |
| Subject2(F2) \_Without Mask | Female subject pronounced the vowel and Empty Data using Without Mask | 300 |

**METHODOLOGICAL INFORMATION**

1. **Experimental Setup Using USRP**

### Description of methods used for collection/generation of data:

The dataset is made up of various actions gathered by the UWB Radar sensor (Xethru X4M03). The Radar is based on Novelda's X4 system-on-chip (SoC), which includes an integrated receiver and transmitter antenna and delivers exceptionally accurate distance and movement measurements. The target was 0.45 metres away from the Radar when the data was collected. Each task took 6 seconds to complete. Through the modular connector XEP, Radar was connected to a PC with Intel(R) Core (TM) i7-7700 3.60 GHz processors and 16 GB RAM. The experimental setup that was used to acquire the data (see Figure 4).

|  |  |
| --- | --- |
|  |  |
| Graphical user interface, application, PowerPoint  Description automatically generated |  |

Figure 4 Experimental Setup

### Methods for processing the data:

Run the Embebbed Software for the X4MO3 development kit from the Xthru Embedded Platform (XEP). The XEP protocol was used to connect the radar transmitter and reception units. Radar operation with XEP requires a system parameter (see Table 2).

Table 3 System Parameters

|  |  |
| --- | --- |
| Parameter | Value |
| Radar Platform | Xethru radar X4M03 |
| Instrumental Range | 9.6m |
| Operating Frequency | 7.29 GHz |
| Transmitter Power | 6.3 dBm |

XEP is free and open-source software. This example is modified to include the Radar as the transmitting and receiving devices. The raw form of the channel state information output is written to .dat files which are then converted to a processable format (see Figure 5).

Diagram

Description automatically generated

Figure 5 Data flow in the data collection stage

Secondly, the radar chip was configured via the XEP interface with x4driver. Data were recorded from the module at 500 frames per second (FPS) in the form of the float message data. A loop was used to read the “Digital Audio Tape” (.dat) data file and save the data into a DataStream variable, which was mapped into a complex range-time-intensity matrix. Thereafter, moving target indication (MTI) filter was applied to get the Doppler range map. Afterwards, the second MTI was used as a Butterworth 4th order filter to generate the Spectrograms using the following parameters: window length, overlap percentage, and fast Fourier transform (FFT) padding factor. In particular, a window length of 128 samples, and a padding factor of 16 was used. In addition, a range profile was created by first converting each chirp to an FFT. A second FFT is then conducted on a defined number of consecutive chirps for a given range bin. Further more, a short time Fourier transform (STFT) was used to create these spectrograms.Spectrogram came into Joint Photographic Experts Group(.jpg) format. The .jpg files would hold the data sets that will be used for training and testing of the Deep Learning (DL) algorithm, see an example in (see Figure 6). The above process was repeated for all the data files in this data set.

Graphical user interface, application

Description automatically generatedGraphical user interface, table

Description automatically generated

Figure 6 Channel state information capture of “A” activity in ".dat" format and the corresponding plot

### Instrument- or software-specific information needed to interpret the data:

Data files are in “. dat” format and convert it into “. Png” using MATLAB Script.